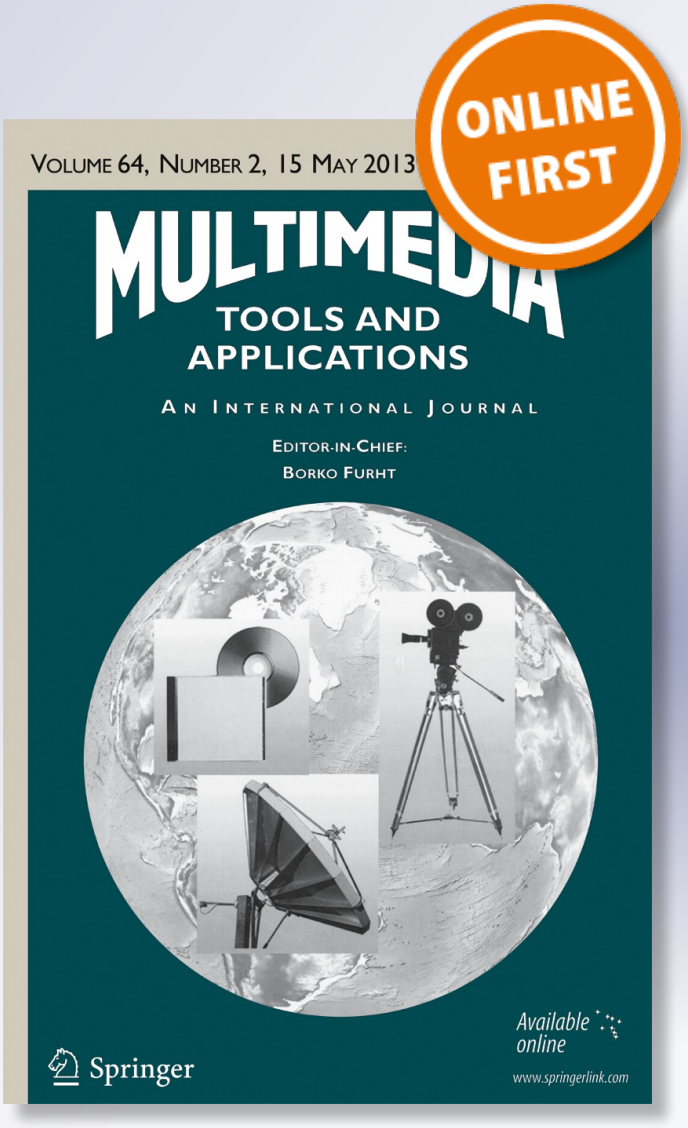
A framework for automatic semantic video annotation

Amjad Altadmri & Amr Ahmed

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-013-1363-6



Springer

Multimed Tools Appl
DOI 10.1007/s11042-013-1363-6

A framework for automatic semantic video annotation

Utilizing similarity and commonsense knowledge bases

Amjad Altadmri · Amr Ahmed

Abstract The rapidly increasing quantity of publicly available videos has driven research into developing automatic tools for indexing, rating, searching and retrieval. Textual semantic representations, such as tagging, labelling and annotation, are often important factors in the process of indexing any video, because of their user-friendly way of representing the semantics appropriate for search and retrieval. Ideally, this annotation should be inspired by the human cognitive way of perceiving and of describing videos. The difference between the low-level visual contents and the corresponding human perception is referred to as the *'semantic gap'*. Tackling this gap is even harder in the case of unconstrained videos, mainly due to the lack of any previous information about the analyzed video on the one hand, and the huge amount of generic knowledge required on the other. This paper introduces a framework for the Automatic Semantic Annotation of unconstrained videos. The proposed framework utilizes two non-domain-specific layers: low-level visual similarity matching, and an annotation analysis that employs commonsense knowledgebases. Commonsense ontology is created by incorporating multiple-structured semantic relationships. Experiments and black-box tests are carried out on standard video databases for action recognition and video information retrieval. White-box tests examine the performance of the individual intermediate layers of the framework, and the evaluation of the results and the statistical analysis show that integrating visual similarity matching with commonsense semantic relationships provides an effective approach to automated video annotation.

Keywords Semantic video annotation · Video search engine · Video information retrieval · Commonsense knowledgebases · Semantic gap

A. Altadmri (✉) · A. Ahmed
School of Computer Science, University of Lincoln, Lincoln, UK
e-mail: atadmri@lincoln.ac.uk

A. Ahmed
e-mail: aahmed@lincoln.ac.uk

Published online: 28 March 2013

1 Introduction

The rapidly increasing quantity of publicly available video data stimulates research into automatic tools for rating, indexing, searching and retrieval purposes. One popular approach is to link a bag-of-words of low-level visual features to each of the identified concepts. Not only does this approach recognize the concepts for which it has been developed, but also, such techniques depend on low-level visual features. Human beings on the other hand utilize a rich descriptive repertoire including details of objects, scenes and activities, and the relationships between these various elements. This gap between human perception and the low-level visual features is referred to as the *'semantic gap'* [6].

In this paper, a framework is presented for a way of video annotation utilizing commonsense human knowledge, embedded in linguistic knowledgebases, in order to address this 'semantic gap' issue. The framework's algorithm detects simple spatio-temporal features in a new query video. These low-level spatio-temporal features are used to find closely-matching videos from a limited annotated database. The annotations of the matching videos are then analyzed and fused taking into account the semantic relationships between the terms encoded in the commonsense knowledgebase, thus resulting in a meaningful annotation for each new video. This annotation is meant to be on the objects, actions and scenes levels, motivated by the fact that human beings annotate videos based on semantic events [5] and context, and not just on the presence of objects.

Why combines visual similarity with ontologies? The idea behind the proposed framework is that in order to bridge the 'semantic gap', both the low-level features and the high-level knowledge have to be utilized in a way that is inspired by human perception. This means that each of these levels has to contribute to the process in each specific area.

For example, the piece of text 'tree', can be considered as equal to an image of a tree, but in a different space. Applying this idea to unconstrained videos leads to the following process: firstly, low-level visual features come to be utilized in finding related nodes in the visual space; and then, real-life high-level semantic knowledge validates these connections in the textual space, so as to come to give a reasonable level of information about the video's contents.

This approach has several contributions to make: firstly, the framework tackles the meaningful annotation of non-domain-specific videos, whereas most of the previous work has been designed to deal with domain-specific videos; secondly, it uses commonsense knowledgebases in this context, including the automated combination of information from ConceptNet and WordNet, and addresses the variation in language used to describe the same entities; thirdly, it analyses sentences to identify the triplet of 'objects, activities and scenes', in a way that is inspired by the human perception, and then utilizes this in correspondence to the visual elements.

Experimental evaluation has been carried out on standard video databases; namely UCF Actions [27] and TRECVID 2005 BBC Rush [38], and has been benchmarked against popular methods of video annotation and high-level concept detection. White-box tests have also been used to examine the individual performance of each of the intermediate layers of the framework. The results show

the effectiveness of the proposed framework in finding semantically representative annotations for the new input video.

The rest of the paper is organized as follows: in Section 2, the key related work is discussed; the proposed framework is presented in Section 3; while the experiments, results and evaluation are described in Section 4; the paper presents the conclusions in Section 5.

2 Related work

A key issue in video annotation is the extraction of reasonably compact features that are representative not just of particular objects, but also of particular actions. A simple approach is to ignore the temporal aspect and to extract visual features from key frames [41]. Alternatively, explicit representations of actions may be sought, for example, state machines may be used to represent spatial transitions of specific objects in a video clip [19], thus regarding an event as a sequence of defined activities. However, this relies on a sophisticated multi-layer system that includes object detection and classification, motion analysis, and motion-blob verification. This means that a vast amount of knowledge is needed in the lower layers, so that this approach is only effective in a domain-specific area, and thus is unsuitable for domain-independent videos.

Good results have been achieved in approaches where some low-level motion information is included. For example, [12] proposed a content-based retrieval system utilizing key frames augmented with a motion histogram. Basharat et al. [7] have developed a generic technique for spatio-temporal feature-extraction and -matching which is relatively simple and robust. This method extracts 3D spatio-temporal volumes (2D spatial and the temporal), and matches these volumes for retrieval. These approaches achieve considerable results in a generic way that is suitable for the domain-independent area. But they are mainly retrieval approaches, which index the low-level features. So, an example of the searched scene or action has to be provided by the end-user.

Recent approaches have made good progress towards tagging videos by searching for near-duplicates. Siersdorfer et al. [34] detects redundant clips to assign new tags. Web near-duplicates videos are explored for data-driven annotation in [44], this showing success in classifying videos when a huge quantity of tagged web videos is freely accessible. Other approaches employ machine learning to produce tags or concepts for visual scenes. Ulges [40] built concept detectors for on-line video tagging. Farhadi et al. [14] introduced a system that computes a score associating a sentence with a query image. A Bayesian model for human action understanding used for video interpretation is presented in [18]. Nearest-Neighbour models with a bag-of-words approach is used for tagging objects in images in [17]. Machine Learning has been effective in most of the domain-specific applications. However, in domain-independent situations, training and class-imbalance become real issues [33].

However, many approaches have tried to utilize ontology in action detection. 'Ontology' is a theoretical representation model in a knowledge system [11]. In [20], an ontology was built by learning relationships between the concepts by analyzing co-occurrences. Other approaches have included visual knowledge directly in multimedia domain-specific ontologies, in the form of low-level visual descriptors

for concept instances, in order to perform semantic annotation [6]. These methods explicitly encode domain knowledge defined by domain-experts, but this makes them impractical for wide domain analysis, and subject to individual design bias.

Commonsense knowledgebases attempt to encode the information and facts that are expected to be known by ordinary people. Well-known commonsense knowledgebases include WordNet [15], *Cyc* [25] and ConceptNet [26]. Currently, ConceptNet is the largest commonsense database built from freely entered text. It is very rich in relationships, the number of assertions and the types of relationships. Commonsense knowledgebases have recently been a popular focus in research into semantic problems. In [43], a trial has started to learn the concept relationships in public video databases which depend on ConceptNet. This trial is mainly to enhance search results of textual queries by way of retrieving the videos for the query and related key search. *ImageNet* [13] is an approach linking WordNet's synsets to images which provides a base for finding representative images for a given a query text.

In [3], an automated enhancement approach to manual annotations is introduced for retrieval purposes, while in [4] a new knowledgebase, *VisualNet*, is proposed. *VisualNet*, which automatically fuses WordNet and ConceptNet, has been developed to serve as a tool for a wide range of visual applications, such as automatic/semi-automatic annotation, indexing, rating and retrieval for both images and videos.

In this paper, the *VisualNet* approach is combined with a visual similarity layer, utilizing spatio-temporal features, to produce an annotation framework suitable for unconstrained videos. A preliminary version of this work was published in conference-form in [2]. This consolidated version is extended by enhancing the first layer, introducing more technical details, and by performing more experiments on common public databases, with deeper analysis and evaluation, using standard TRECVID measures.

3 Proposed framework

Two main issues in semantic annotation of unconstrained videos emerge from the discussion in Sections 1 and 2: firstly, the extraction of a compact representation composed of spatio-temporal features, suitable for efficient matching of objects, actions and scenes; and secondly, the representation and use of semantic relationships between objects, actions and scenes to validate annotation, compensating for the limitations of the raw visual information, including variable appearance, occlusions and ambiguity. This motivates the structure of the proposed framework, which is depicted in Fig. 1.

The input is a *query video* to be annotated (e.g. for indexing or retrieval purposes): the output is an annotation that semantically represents the objects, actions and scene in the query video, illustrated in Fig. 2.

As depicted in Figs. 1 and 2, the proposed framework consists of two layers. In the first layer, an initial weighted list of potential free-text annotations for the query video is obtained. This is done by comparing the dominant low-level spatio-temporal features from the query video with the corresponding features of videos from a pre-annotated dataset, and by selecting the annotations of the most closely-matched videos. In the second layer, the selected potential annotations are analyzed in order to consolidate the annotation of the query video by exploiting the

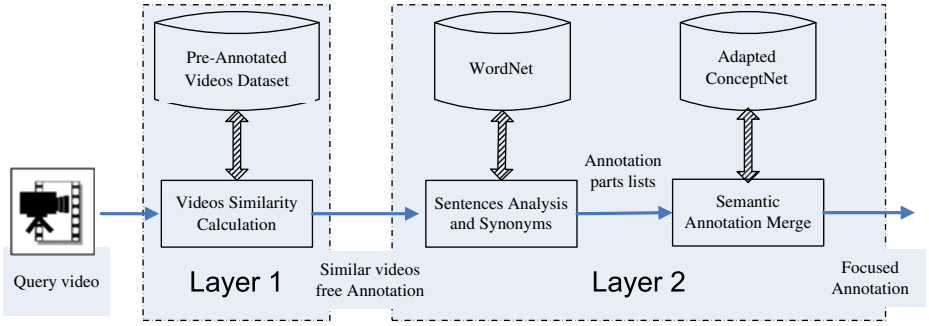


Fig. 1 The automatic semantic annotation framework. The input is a query video and the output is the annotation. Layer 1 finds a list of visually similar videos; Layer 2 processes their annotations semantically to produce the final annotation. Layer 2 is composed of two layers: the first analyses the initial annotations and finds commonalities, the second validates the consistency of the parts of the sentence

semantic relationships between the terms used in the retrieved annotations. The next subsections describe the framework in detail.

3.1 Layer 1: visual similarity

In this layer, a generic non-domain-specific similarity calculation method is utilized to detect the most similar videos to the query video from a dataset. Briefly, the keypoints are identified and tracked to construct 'keypoint trajectories' in each video file (Section 3.1.1). Then a video signature is extracted out of these trajectories for this video (Section 3.1.2). In the comparison phase, i.e. the distance measured between the query and each dataset's files, the video signatures are compared, and

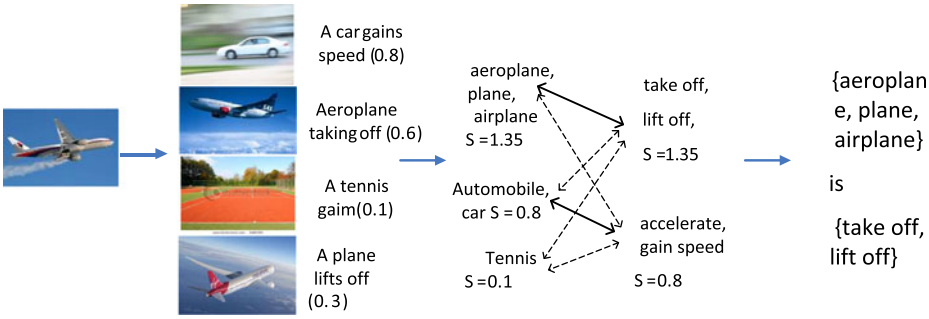


Fig. 2 An example for the framework: an airplane is taking off. The first similar video retrieved is a false positive as it is a car. Similarly, the third one is a false positive, and it also contains a misspelling 'gaim' instead of 'game'. The example shows how the annotations of the correct retrievals are grouped together semantically. The calculation for first 'aeroplane' weighing S = 1.35 comes from the calculation S = (0.6 + 0.5*0.3) + (0.3 + 0.5*0.6) = 1.5 * (0.3 + 0.6). This is because 0.5 * 0.3 processing the synonym 'plane', and then the opposite calculation for 'plane' (now treating 'aeroplane' as the synonym), then summing the results

a consolidated similarity-measure is calculated between the query video and each video in the database (Section 3.1.3).

The following subsections present the details of this layer's components.

3.1.1 Trajectories calculation

Interest points are identified in each frame using Lowe's implementation of SIFT features [28]. Then, temporal trajectories across all the frames are built by connecting matched interest points between successive frames. Interest points are matched by extracting a 128-dimension feature vector, and calculating the distance using (1):

$$\theta_{i,j} = arcos \frac{\mathbf{x}_{i,t} \cdot \mathbf{x}_{j,t+1}}{|\mathbf{x}_{i,t}||\mathbf{x}_{j,t+1}|} \tag{1}$$

where $\mathbf{x}_{i,t}$ and $\mathbf{x}_{j,t+1}$ are the vectors representing interest points i and j from frame t and frame $t+1$ respectively, and θ is the angle between $\mathbf{x}_{i,t}$, $\mathbf{x}_{j,t+1}$, which represents the similarity distance.

An interest point is matched to the most similar point (in feature space) in the next frame, provided that the feature-distance is significantly less than that of the next nearest one. This condition is satisfied by the formula in (2):

$$\theta_{i,j1} < \beta\theta_{i,j2} \tag{2}$$

where β is a coefficient determining how much nearer, in feature space, $\mathbf{x}_{i,t}$ must be to $\mathbf{x}_{j1,t+1}$ than to $\mathbf{x}_{j2,t+1}$ to be a potential match, and $j1$ and $j2$ denote the closest and second-closest matches respectively. β is set to 0.6 based on [8]. To ensure that matching is one-to-one, the condition is also applied in reverse; that is, the distances from $\mathbf{x}_{j1,t+1}$ to all points in the previous frame are calculated, using (1), to ensure that $\mathbf{x}_{i,t}$ is the nearest match to $\mathbf{x}_{j1,t+1}$, and is at least β times closer than the next nearest point [8].

The final check is performed to verify that the spatial positions of the interest points in the frames do not exceed a maximum distance apart. The formula in (3) satisfies this condition:

$$||\mathbf{q}_{i,t} - \mathbf{q}_{j,t+1}|| < \alpha \tag{3}$$

where $\mathbf{q}$ denotes the frame position of the interest point, and $\alpha = 30$ is the distance threshold in pixels, which is empirically selected to guarantee the smoothness of the objects' transition between successive frames, according to the smooth movement assumption within single video shot [1] (i.e. no sudden unrealistic movement). It has been selected using experiments on random video clips and it is not tuned in the experiments according to the test data.

The resulting trajectories are post-processed as follows: firstly, 'broken' trajectories are repaired by merging them if there is a small frame gap between the two ends, provided that the ends match using the conditions described above ((2) and (3)); secondly, short trajectories are eliminated, including singleton keypoints. The resulting trajectories are then used to extract the video signature, as explained in the next subsection.

3.1.2 Video signature

The trajectories extracted in the previous phase are processed to provide a video signature, which is a compact representation of spatio-temporal information that is suitable for comparison. Each trajectory is represented by two values: the 128D feature vector at the mid-point of the trajectory, $\mathbf{t}_{i,m}$ (the m^{th} trajectory of the i^{th} video) and a weight representing the normalized frame length of this trajectory, $w_{i,m}$. The weights are normalized to lie in the range [1, Q], as demonstrated in (4).

$$w_{i,m} = (Q-1)\frac{L_{i,m} - \min_L^i}{\max_L^i - \min_L^i} + 1 \tag{4}$$

where $L_{i,m}$ is the length of the trajectory's frames, and $\min_L^i$ and $\max_L^i$ are the minimum and the maximum lengths, respectively, for all trajectories in the i^{th} video. Q is the maximum targeted value for normalization, which starts from 1 to not lose the shortest trajectories.

Representing interest point trajectories in this fashion adds the simplest temporal information (relative duration) to the SIFT features, but, nevertheless gives a better performance than simply matching interest points from key frames, as it concentrates on matching important persistent points, and reduces the effect of noise, as presented in Section 4.2.

More complex methods and features (such as bag-of-words) could potentially be used in this layer. However, a simple approach is used, as the focus is on the contribution of the framework and its ability to annotate videos, even without having the best results from the first layer.

3.1.3 Video matching

In this phase, the extracted signature of the query video i is compared against the signature of each video j from the database to find the most similar videos.

The distance between the two videos' signatures is calculated as follows: after matching the trajectories, using (1) and (2), the weights of these matched trajectory pairs are summed, and then normalized over the total number of trajectories, as depicted in (5):

$$s_{i,j} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} I_{m,n}(w_{i,m} + w_{j,n})}{\min(M, N)} \tag{5}$$

where $s_{i,j}$ is the similarity degree between query video i and database video j, M and N are the number of trajectories respectively in each of these videos, and $I_{m,n}$ is a binary indicator variable with value $I_{m,n} = 1$ if the trajectories match according to (1) and (2), and value $I_{m,n} = 0$ otherwise.

The database's videos are sorted based on the similarity degree, $s_{i,j}$, and the most highly ranked videos are selected for annotation analysis.

3.2 Layer 2: knowledge processing

The second layer uses the commonsense knowledgebases to derive annotations from similar database videos, by looking for semantically consistent and coherent terms from among the annotations of these videos. It exploits relationships between

the three elements: objects, actions and scenes. As depicted in Fig. 1, this layer consists of two stages; *sentence analysis* and *semantic annotation merge*. The following subsections discuss these stages.

3.2.1 Sentence analysis

This stage focuses on analyzing the potential annotation sentences, obtained from Layer 1 (Section 3.1), and on finding annotations with similar semantic meanings that represents the input video. This accounts for the variety of alternative names used for the same or similar objects (e.g. car, automobile), of descriptions of events or actions (e.g. put, set, place), or different spellings (e.g. 'aeroplane' in British, 'airplane' in American English).

If the database videos are annotated using free text containing full or semi-sentences, then firstly each annotation is divided into an Object, Action and Scene triplet. The Stanford NLP Log-linear Part-Of-Speech Tagger [37] is used to obtain the parts of the sentence. These tags indicate which part is the object (the subject in linguistic terminology), which is the action (the verb and its related prepositions), and which is the scene (the location), if it exists.

Three separate lists are generated from this analysis. The *Objects* and *Scenes* lists contain *nouns*, whereas the *Actions* list contains *verbs*. This helps to prevent confusion over words which have multiple meanings as both verbs and nouns (e.g. 'fly' which could refer to an 'insect', noun, or the action of flying, verb).

Each entry in these lists is returned to its primary form that matches the list's type, using WordNet [15] 'baseForm' function (e.g. Action: 'looking' ⇒ 'look'). Synonyms are added with a weight w_s, which is calculated from the initial word weight w_i, as follows:

$$w_s = w_i \times d \tag{6}$$

where $d \in [0, 1]$ is a constant that ensures assigning a lower weight to synonyms than the original word, $d = 0.5$ is selected empirically. Values below 0.3 tend to make the step meaningless, while values above 0.8 tend to increase the false-positive rate.

As each entry contains the primary form of multiple synonyms, as explained earlier, repeated entries in each list are merged and their weights accumulated. Then, the weights of the resulting lists are normalized so that the largest weight in the list is equal to 1.0, using (7):

$$w'_k = \frac{w_k - \min_w}{\max_w - \min_w} \tag{7}$$

where w_k and w'_k are the original and normalized weights, respectively, for an entry k; $\min_w$ and $\max_w$ are the maximum and minimum weights for the whole list entries.

The example in Fig. 2 illustrates the previous steps by using a query video of an airplane taking off. There are two videos retrieved, as similar, that are false-positives containing a car and a tennis game with similarity weights equal to 0.8 and 0.1, respectively. Two true-positive videos, with weights 0.6 and 0.3 respectively, were retrieved. The calculations in Layer 2 are performed using the previous equations, for example, the synonym set {take off, lift off} is created twice, one via 'take off' and the other via 'lift off'. The total weight, which is 1.35 before normalizing, is calculated as S = (0.6 + 0.5*0.3) + (0.3 + 0.5*0.6) = 1.5 * (0.3 + 0.6), where 0.5 is the d constant in (6). It can be noted that the misspelled word 'gaim' has been eliminated.

At the end of this stage, the output amounts to three sorted lists, each of which contains weighted entries for one part of the scene elements (object, action and background scene or context). The benefits of this process are:

1. repeated matched elements are consolidated with high weighting; for example, if two videos are matched, one annotated with 'car speeding up' and the other with 'car braking', it is concluded that the video includes a car regardless of the action; similarly, having two annotations, 'boat sailing at sea' and 'plane landing on the sea', both identify the scene background as the 'sea'.
2. synonyms such as 'car', 'auto' and 'automobile' are detected and grouped with a higher confidence weight.
3. different spellings in different languages (e.g. 'armored' and 'armoured') are detected and consolidated.
4. mis-spelled words, such as *'gaim'* in Fig. 2, are ignored, and special annotations and proper names are down-weighted by the nature of the process, as they do not generally occur multiple times (e.g. '*Jack* running').

The sorted lists will be exploited and analyzed further to produce appropriate semantic annotation for the input video, as explained in the following subsection.

3.2.2 Semantic annotation merge

The aim of this stage is to exploit the semantic relationships between the different terms in the three lists, giving higher weighting to semantically associated terms and to discover plausible relationships between the sentences' parts. For this stage, ConceptNet [26] is adopted, filtered and modified, which contains a huge number of concept nodes. Each concept is a semi-sentence or a phrase. The rest of this subsection gives a concise description of ConceptNet, and explains how it has been automatically adapted for this purpose, and then explains the annotation composition selected using this adapted version.

ConceptNet ConceptNet is currently considered to be the largest of all commonsense knowledgebases [21, 26]. Each node is a concept, which is in itself a part of a sentence expressing a meaning. ConceptNet is a very rich knowledgebase in several ways: firstly, the huge number of assertions and nodes contained; secondly, the wide range of information included; and finally, the various types of existing relationships that hold description parameters. Figure 3 presents a snapshot of ConceptNet. In

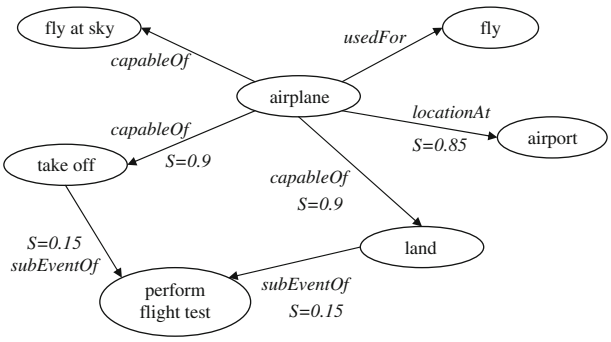


Fig. 3 A snapshot of ConceptNet relationships

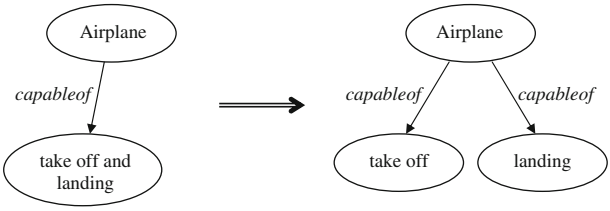


Fig. 4 An example of nodes analysis. The compound concept node 'take off and landing' is divided into two nodes 'take off' and 'landing'.

contrast to WordNet, ConceptNet is very useful in describing real life scenes, but it is weak in identifying the exact relation between related-meaning words.

ConceptNet adaptation Firstly, the relationships' types that are most useful in the visual field are selected. These relationships are: *capableOf*, *usedFor* and *locationAt*. The *capableOf* and *usedFor* relationships are merged into one, called *does*, by adding the scores of matched relations in both lists.

The *does* and *locationAt* relationships are used to connect the parts of the annotation as follows: the *does* relationship shows whether a specific action could be associated with a specific object; and the *locationAt* relationship indicates whether a scene may be associated with an object or action.

In many text mining applications, the phrases of ConceptNet's nodes are more efficiently used in a direct way. In contrast, in this case, the aim is to form a meaning that simulates the triplet of the visual world - objects, scenes and actions. ConceptNet's nodes also need to be in a format that enables comparison with WordNet's nodes. To achieve this, each ConceptNet node is analyzed automatically to obtain visually relevant terms. The rest of the node is then deleted.

This modification process is performed as follows: firstly, the words of each node are tagged using the Stanford tagger [37]; parts of the sentence that are not useful in the visual field are then deleted, (these parts include some prepositions, stop words and some commonly used adjectives and adverbs, which are held in an exclusion table. For example, 'fast' is a useful visual adjective because it conveys a meaning related to motion, but 'better' is not.); finally, a split operation is applied in order to divide the composite nodes, producing new relationships. An example is illustrated in Figs. 4 and 5.

Annotation composition The result of the previous process is a filtered version of ConceptNet, with a new structure more suitable for matching with the parts of visual

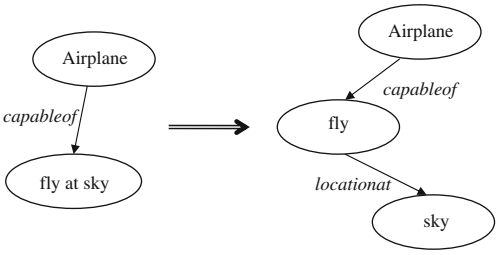


Fig. 5 An example of nodes analysis. The compound concept node 'fly at sky' actually has two separate elements; the *action* and *scene*. Consequently, the relationship is split

scenes. As the *does* relationship refers to the possibility of assigning an action to an object, the objects list and the actions list are cross-validated using this relationship; formulized in (8).

$$T = \{r \in R : \forall o \in O, \forall a \in A, \exists r \in R_{\text{does}}, core(r_{\text{start}}) \cap o \neq \emptyset, core(r_{\text{end}}) \cap a \neq \emptyset\} \tag{8}$$

where T is the resulted sentences' set, R and R_{does} are two sets that represent all ConceptNet's relationships and their *does* subset respectively, and O and A are the *Objects* and *Actions* lists respectively. And $core(NODE)$ is the proposed function described in previous paragraph that extract the meaning core of a ConceptNet node. Finally, r_{start} and r_{end} are the start and end nodes of the relation r respectively.

The cross-weight for each sentence t in the set T is calculated based on the weight of its parts and the weight of the validation relationship, as in (9).

$$W_t = w'_o w'_a s_r \tag{9}$$

where W_t is the sentence weight, w'_o and w'_a are the normalized weights of noun and verb phrases in objects and actions lists, respectively, resulting from (7), and s_r is the relationship score obtained from the proposed adapted version of the ConceptNet.

Then, the same operation is performed between the objects lists and the scenes lists, using the *locationAt* relation.

All the previous matching of textual terms has been performed at the stemmed words level. This is done by stemming all the words of each entry, that is obtaining the root of the word, then sorting the resulting stemmed words alphabetically. This removes arbitrary differences caused by the choice of phraseology; for example, 'seasonal flowers' and 'flowers in season' both become 'flower season'.

This stage represents the last step of the proposed framework that produces the final output. This output is a weighted list of candidate annotation sentences for the query video, as seen in Figs. 1 and 2. The next section details the experiments and the evaluation of the results.

4 Experiments, results and evaluation

To evaluate the framework, various experiments were carried out over a couple of standard datasets. Standard evaluation measures, from TRECVID [32], have also been calculated to benchmark with the two selected baselines. These standard measures are selected to evaluate black-box and white-box tests on the framework. The black-box tests include benchmarking the performance precision of the proposed framework's final output against the baselines. Meanwhile, the white-box tests examine the effectiveness of the intermediate stages. Layer 1 is evaluated against the selected baselines as a categorization problem, and confusion matrices are calculated. In addition to this, a statistical analysis has been carried out regarding these results in order to consolidate the evaluation, and ConceptNet's nodes and relations selection is also examined.

Experiments have been performed on two standard datasets for action recognition and video information retrieval; UCF Actions [27, 39], and TRECVID BBC Rush

[38]. The first one includes various activities involving humans and animals in different indoor and outdoor activities, while the second contains many types of man-made objects in various manoeuvring scenarios designed for information retrieval.

These challenging videos contain a considerable range of variations including types of objects and actions, as well as size, appearance, shape, viewpoint and motion of objects. In addition, there are variable camera quality and motion issues, including pan, tilt and zoom. All these challenges combined complicate the annotation task considerably.

To benchmark the framework's performance, two baselines have been selected, as follows. The first one is following annotation by search, where each video is presented by visual features selected from its key frame, and then the annotations of the videos with the nearest features' vectors are assigned weights that reflect their similarity distance from the query video. The key frames are extracted based on [36]. This selected baseline's features are the interest points attributed by 128D SIFT features. This one is selected as it proved to perform best in many applications such as recognition and retrieval in many evaluation studies, as in [10, 16, 24, 35].

The second baseline is based on high-level concept detection. LSCOM semantic concepts detectors are selected based on VIREO-374 bag-of-visual-word features [23]. This approach is chosen as it has been widely used as baseline recently [30, 42].

In the rest of this section, the experiments involving the two selected datasets are detailed.

4.1 Black-box tests

These black-box tests benchmark the precision performance of the framework's final output against the baselines.

4.1.1 UCF actions dataset

The first experiment is performed on the UCF Actions [27], which can be downloaded from [39]. This dataset contains 1600 video clips related to many activities including: basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking a dog. A thumbnail sample of this dataset is presented in Fig. 6.

The dataset is organized in such a way that visually similar videos, within each category, have been grouped together. Each main category contains 25 subgroups with about six action clips in each. The clips within each subgroup may share some features such as similar background or a similar actor. However, they may have different viewpoints, directions of motion/action. This dataset is very challenging as it contains large variations in object appearance, pose and scale, as well as the camera's movement, viewpoint, illumination conditions and cluttered background.

Experiment For experimental purposes, one video clip has been randomly taken from each of the subgroups to form the test dataset. The rest are used as the pre-annotated dataset. As a result of this, the test set contains 275 videos whilst the indexed set contains 1,324 videos (one file was found to be corrupted, hence excluded).

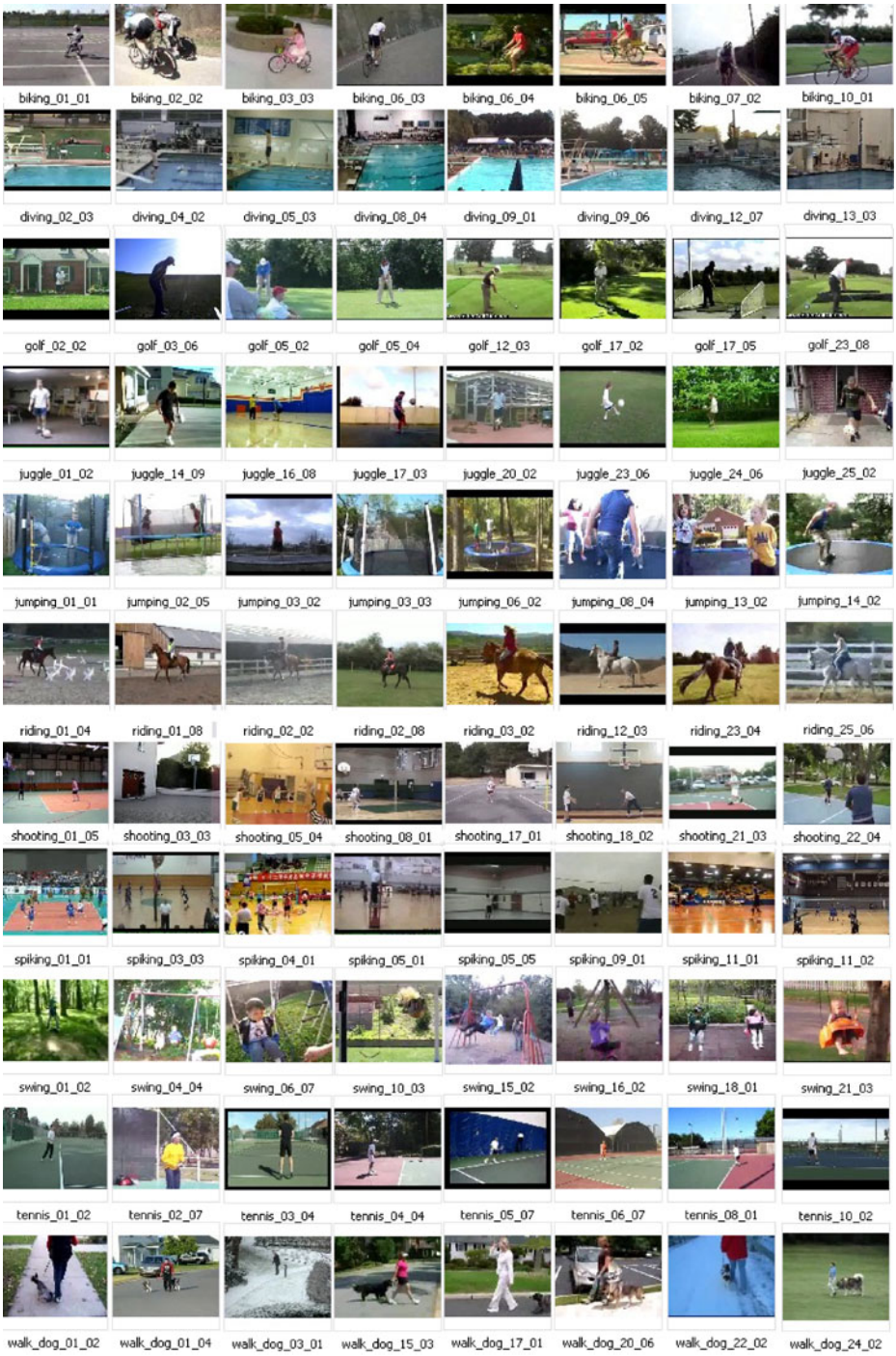


Fig. 6 A UCF actions videos dataset snapshot. Each row illustrates eight thumbnails from one of the 11 categories. The random thumbnails show the huge variations of visual features internally within each category, and externally between all categories

The framework is evaluated against the baselines as an annotation problem. Each annotation sentence retrieved is divided into its basic textual terms, where each term is compared separately to the ground truth pool.

For the framework evaluation, the performance of the proposed method against the baseline is compared, using the precision considering different number of ranked files, which is a TRECVID standard measure [32]. The precision at a given cut-off rank is defined in (10):

$$P(r) = \frac{R_{1-r}}{r} \tag{10}$$

where r is the rank studied, R_{1-r} is the number of retrieved annotation terms of rank r, or less, that are relevant, i.e. describe a part of the scene.

Figure 7 presents the resulting $P(r)$ for the framework against the baselines over various cut-off ranks. The figure shows that the proposed framework consistently outperforms the baselines. We obtain a precision of 0.80 at the first annotation, dropping to 0.5 at the ninth one; while the SIFT baseline starts at 0.59 for the first annotation, dropping to 0.31 at the ninth. The LSCOM baseline drops from 0.67 to 0.38 in the same range.

Figure 8 illustrates a snapshot of the qualitative evaluation of the query videos' test in the experiments with the two datasets. For each example, it shows a sample of the produced annotations, where each annotation is labelled with either a *noun* or *verb* and associated with its weight. The column on the far right shows a sample of the composed annotations. The underlined ones are the correct relevant ones, while the others are false alarm.

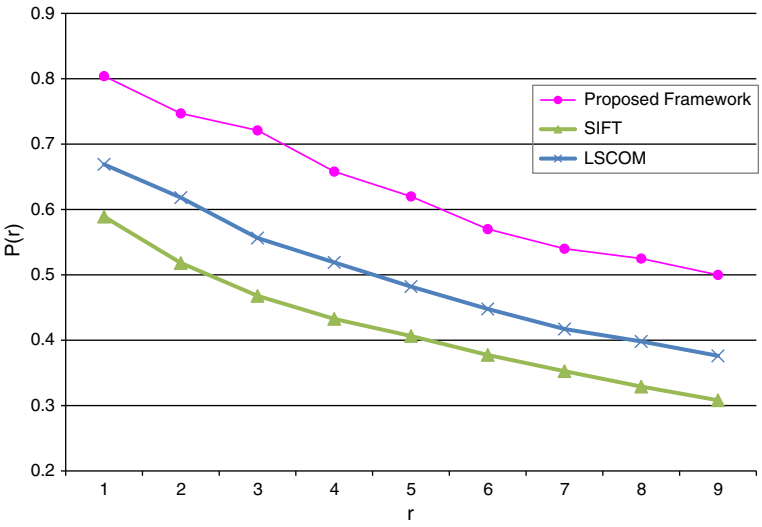


Fig. 7 Precision at a given cut-off rank for UCF Actions dataset experiment. The proposed framework shows considerable improvement over the baseline all over the curve. Even the number of false alarms increases as more output items are considered; however, items with a high rank are more important

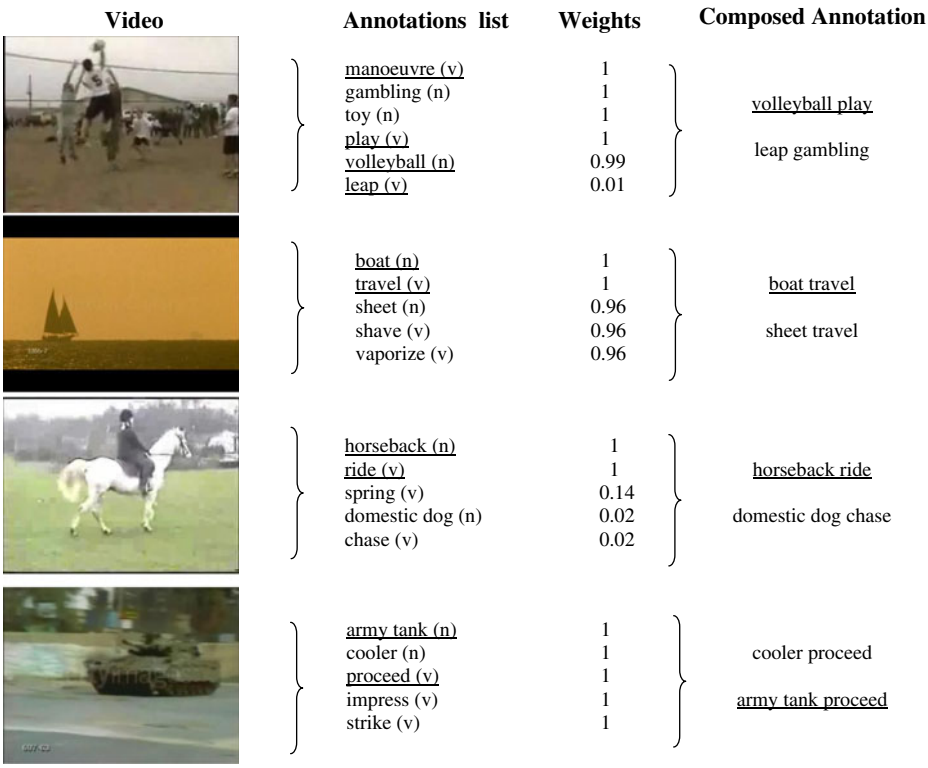


Fig. 8 A snapshot of the output of the framework, represented as parts-of-speech, and composite connections, as sentence cores. The underlined results are the correct ones, while the others are false alarms. NLP tools could be utilized to form a refinement for sentences adding suitable propositions, etc

4.1.2 TRECVID BBC rush dataset

The second experiment is carried out using TRECVID BBC Rushes [38], a group of standard databases for video information retrieval. This dataset contains 335 single-shot video clips containing various types of moving man-made objects such as cars, tanks, airplanes and boats. A snapshot of some videos in this dataset is illustrated in Fig. 9. Again these pre-recorded videos contain a considerable range of variations, including the objects size, appearance, shape, viewpoint and motion. In addition, there are many possibilities of unknown camera quality and movement.

Experiment For experimental purposes, one video clip from the database is taken each time as a query video to be used against the others as the pre-annotated dataset, and then the rates are averaged. This follows the standard leave-one-out evaluation protocol [9, 22, 31]. Hence, each time, there is one test file and 334 pre-annotated indexed ones.

The framework is evaluated against the baseline as an annotation problem. In a similar way to Section 4.1.1, the performance of the method is compared against the baseline using the precision of the numbers of ranked files (10).

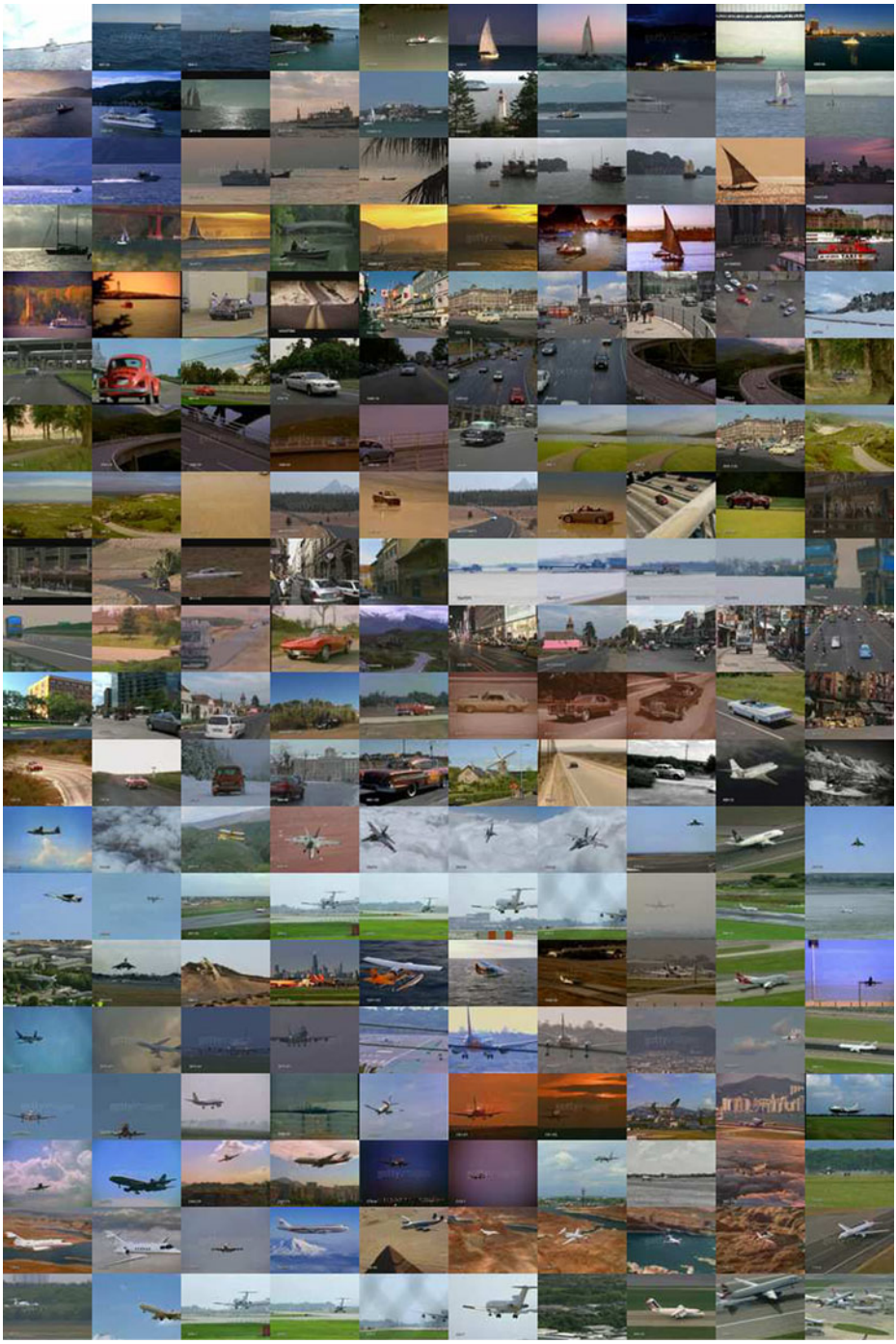


Fig. 9 A BBC Rush dataset snapshot. The random thumbnails show the considerable range of variations either in the visual contents' properties or in the camera quality and movements

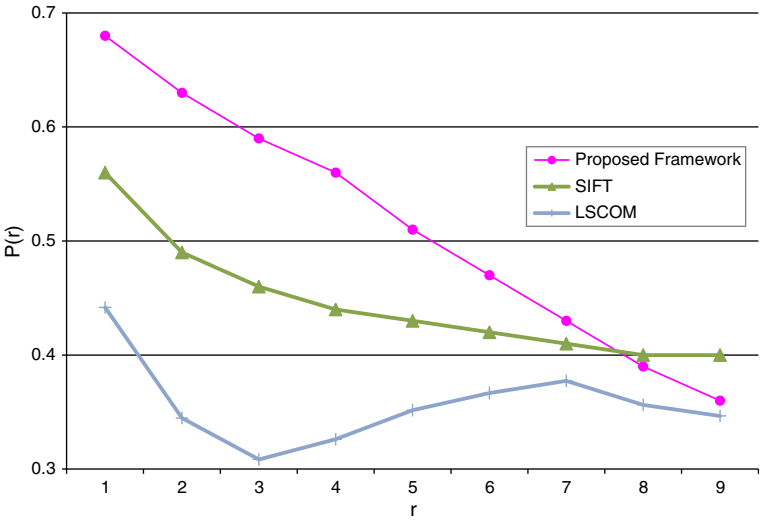


Fig. 10 The precision at a given cut-off rank for BBC Rush dataset experiment. The proposed framework shows a considerable improvement over the baselines when it comes to the high ranks. The performance of the framework comes close to the baselines as more output items are considered, and, further, it goes below the SIFT baseline after the 8th rank. However, items with a high rank are more important

The results of this precision at a given cut-off rank is illustrated in Fig. 10, which demonstrates that the proposed framework outperforms the SIFT baseline over the ranks #1 till #7. The proposed framework attains a precision of 0.68 at the first annotation, dropping to 0.51 at the fifth one and 0.36 at the ninth; whilst this baseline starts at 0.56 for the first annotation, dropping to 0.43 at the fifth, and 0.4 at the ninth.

	spiking	walk_dog	jumping	riding	biking	golf	juggle	swing	tennis	diving	shooting
spiking	0.48	0.04	0.28	0.00	0.00	0.00	0.00	0.04	0.08	0.00	0.08
walk_dog	0.04	0.52	0.32	0.00	0.00	0.04	0.00	0.00	0.04	0.00	0.04
jumping	0.04	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
riding	0.00	0.00	0.28	0.68	0.00	0.00	0.00	0.00	0.04	0.00	0.00
biking	0.04	0.00	0.32	0.00	0.52	0.00	0.00	0.04	0.04	0.00	0.04
golf	0.00	0.00	0.24	0.04	0.00	0.68	0.00	0.00	0.00	0.00	0.04
juggle	0.00	0.00	0.04	0.00	0.00	0.00	0.92	0.00	0.00	0.00	0.04
swing	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.00
tennis	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.96	0.00	0.00
diving	0.00	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.04	0.72	0.00
shooting	0.00	0.00	0.04	0.04	0.00	0.04	0.00	0.00	0.00	0.00	0.88

Fig. 11 The UCF action confusion matrix for the proposed method Layer 1. The *horizontal rows* are the ground truth, whilst the *vertical columns* are the retrieved videos. The *main diagonal line* represents correct classification. Most false positives relate to actions classified as 'jumping'. This is due to the fact that many of them contain 'jumping' as a sub-action

	spiking	walk_dog	jumping	riding	biking	golf	juggle	swing	tennis	diving	shooting
spiking	0.52	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.08	0.04	0.24
walk_dog	0.00	0.20	0.00	0.04	0.04	0.00	0.08	0.00	0.20	0.00	0.44
jumping	0.04	0.00	0.72	0.04	0.00	0.04	0.00	0.00	0.00	0.04	0.12
riding	0.00	0.00	0.00	0.20	0.00	0.28	0.00	0.00	0.08	0.16	0.28
biking	0.08	0.04	0.04	0.00	0.20	0.16	0.04	0.00	0.04	0.08	0.32
golf	0.00	0.00	0.04	0.00	0.08	0.76	0.00	0.00	0.04	0.00	0.08
juggle	0.00	0.00	0.00	0.00	0.00	0.08	0.80	0.00	0.00	0.08	0.04
swing	0.04	0.04	0.00	0.00	0.04	0.00	0.00	0.68	0.12	0.00	0.08
tennis	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.84	0.04	0.08
diving	0.00	0.00	0.00	0.00	0.00	0.16	0.12	0.00	0.04	0.64	0.04
shooting	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.00	0.00	0.92

Fig. 12 The UCF action confusion matrix for the first baseline Layer 1

Moreover, the second baseline shows low variant performance going down sharply from 0.44 precision at the beginning to 0.31 at the third rank to oscillate around 0.35 after.

There are three important points to note: firstly, the first baseline's performance comes close to that of the framework, and then outperforms it at lower ranks. But the ideal, and the more important, results are those that are the top highly ranked ones; secondly, by comparing the performance of Layer 1 in the white-box tests in Section 4.2, Table 2, although the framework shows no significant superiority in the Layer 1 at the Rank 1, Table 2, the whole framework output manages to indicate a significant difference; finally, by comparing with the first experiment, the curve drops more quickly due to the fact that when precision of similarity is decreasing below 50 %, with less variety of video groups, the random outliers tend to share annotation parts; and while Layer 2 gathers the repeated and compatible meanings which resulted from Layer1, more outliers tend to have higher weights as a final output.

4.2 White-box tests

In this section, white-box tests examine the effectiveness of the intermediate stages, namely Layer 1 and the ConceptNet adaptation process, on the whole framework performance. Layer 1 is evaluated against the baseline as a categorization problem.

Table 1 Comparing the two groups (Layer 1 of this framework (G1) and Layer 1 of the baseline (G2)) using the results of the unpaired *t* test based on accuracy

Number of files	G1 mean	G2 mean	G1 SD	G2 SD	P value
1	0.74	0.59	0.44	0.49	0.0001
5	0.54	0.41	0.35	0.38	0.0001
10	0.38	0.29	0.27	0.29	0.0001

SD stands for 'standard deviation'. The analysis is calculated over the 275 test files. By conventional criteria, noticing the two-tailed P value, this difference is considered to be *extremely statistically significant*

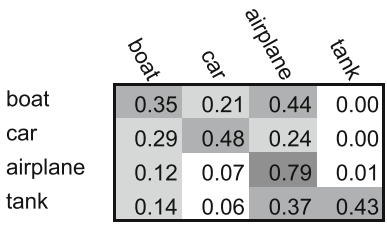


Fig. 13 In the BBC rush confusion matrix for the proposed method Layer 1, the *horizontal rows* are ground truth, whilst the *vertical columns* are retrieved videos. The *main diagonal line* represents the correct classification

Thus, each file retrieved is considered as a false alarm if it belongs to a different category from the query file, based on the ground truth pool.

4.2.1 UCF actions dataset

Figures 11 and 12 illustrate the confusion matrix for Layer 1 in the proposed method and the first baseline, respectively. In this comparison, only the highest-ranked video is considered and evaluated in the previously mentioned way.

It is noticeable that this method assigns a number of false positives to the 'jumping' category. However, on closer inspection, it has been found that they may actually contain 'jumping' as a sub-action. This confirms the improvement of the proposed signature over the key frames only, as the proposed video signature manages to take into account the temporal relationship of the points. Nevertheless, the framework has not been customized to address this particular case, as it has to function with wide domain videos.

From Figs. 11 and 12, the proposed method performs best in eight categories against three categories for the baseline. In addition to this, the overall accuracy of the proposed method is 75 %, compared to 59 % for the baseline, indicating a better overall performance. To confirm and consolidate the evaluation, deeper investigation, including statistical analysis, was carried out using these results. The unpaired t test [29] has been used to compare the results of Layer 1 of the proposed framework (Group 1), and the baseline (Group 2). The analysis is calculated over the 275 test files and the results illustrated in Table 1. In noticing the two-tailed P value, by conventional criteria, this difference is considered to be *extremely statistically significant*. Similarly, the standard deviation for this method is less than that for the baseline considering the different number of files. This indicates that this method is not only significantly more accurate, but is even more precise.

TRECVID BBC rush dataset In this section, Layer 1 is evaluated against the baseline as a categorization problem. Thus, each file retrieved is considered as a false

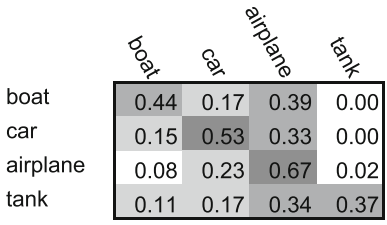


Fig. 14 In the Accuracy Confusion Table for the baseline for BBC Rush dataset, the *horizontal lines* are ground truth, whilst the *vertical columns* are the retrieved files

Table 2 Comparing the two groups (Layer 1 of this framework (G1) and Layer 1 of the baseline (G2)) using the results of the unpaired *t* test based on accuracy

Number of files	G1 mean	G2 mean	G1 SD	G2 SD	P value
1	0.58	0.56	0.49	0.50	0.4834
5	0.50	0.43	0.32	0.33	0.0100
10	0.46	0.40	0.27	0.27	0.0022

SD stands for 'standard deviation'. The analysis is calculated using the 335 test files from the BBC Rush dataset. Although, using the conventional criteria, noticing the two-tailed P value, the difference is considered to be *not statistically significant* at rank #1, but it became *very statistically significant* at ranks #5 and #10

alarm if it belongs to a different category from the query file, based on the ground truth pool.

Layer 1 confusion matrices in this method and the baseline are illustrated in Figs. 13 and 14, respectively. In this comparison, the most highly-ranked video is considered and evaluated in the previously mentioned way.

The proposed method performs better in half of the categories as compared to the baseline, which itself performs better in the other half. However, the overall accuracy of the one proposed is still slightly better in the top ranked file, which is 58 % compared to 56 % for the baseline, Table 2.

In a similar way to the previous experiment, the unpaired *t* test [29] has been applied for deeper statistical analysis. This test compares the results of Layer 1 of the framework (Group 1), and the baseline (Group 2).

The analysis is calculated using the 335 test files, and the results are illustrated in Table 2. Noticing the two-tailed P value, by conventional criteria, even though this

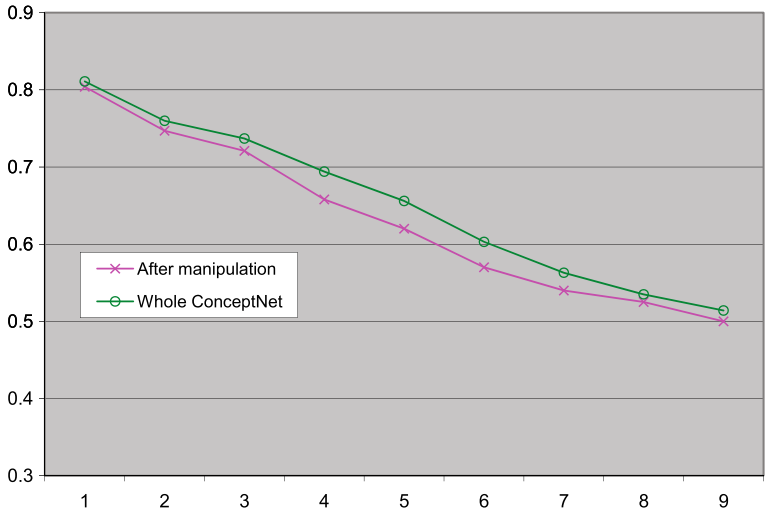


Fig. 15 Accuracy using all ConceptNet's relations in comparison with the selected ones. The adapted version shows a tiny loss of accuracy compared with the original full ones

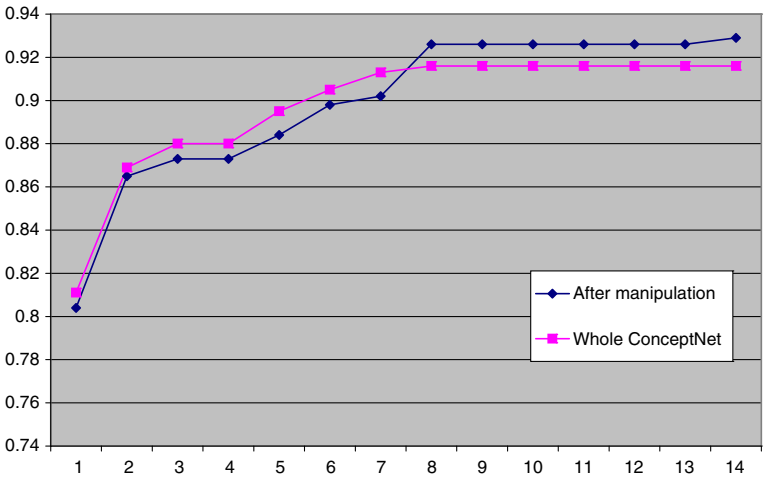


Fig. 16 Precision-over-N (P_N) using all ConceptNet's relations with the selected ones. The original full version will slightly outperform the selected one if the result page contains seven or less items, but will become worse if it contains over seven

difference is not considered to be *statistically significant* at rank #1, it becomes *very statistically significant* at ranks #5 and #10.

4.2.2 ConceptNet's nodes and relations filtering

Another white-box test is performed to investigate the validity of eliminating nodes and relations from ConceptNet, in Section 3.2.2. In this test, the same adaptation operations are performed on all ConceptNet's relations rather than only the visually-related ones. Figures 15 and 16 compare the results of the framework using all relations and nodes of ConceptNet with the selected ones only. The evaluation is based on Accuracy and Precision at (r) from (10), respectively. For deeper investigation, the statistical analysis for the same measures using t test is illustrated in Tables 3 and 4, respectively.

Using conventional criteria, the results of t test considering the difference between the two groups to be *not statistically significant*. In analyzing these figures and statistics tests, it is noticed that there is no significant difference between using either version from the quality point of view. This proves that the suggested process

Table 3 Comparing the two groups (the whole of ConceptNet (G1) and the proposed skimmed version (G2)) using the results of the unpaired t test based on precision

Rank	G1 mean	G2 mean	G1 SD	G2 SD	P value
1	0.80	0.81	0.40	0.39	0.83
5	0.62	0.66	0.32	0.32	0.20
10	0.48	0.50	0.27	0.26	0.48

SD stands for 'standard deviation'. The analysis is calculated using the 275 test files. By conventional criteria, noticing the two-tailed P value, this difference is considered to be *not statistically significant*

Table 4 Comparing the two groups (the whole of ConceptNet (G1) and the proposed skimmed version (G2)) using the results of the unpaired *t* test based on Precision-over-N

Rank	G1 mean	G2 mean	G1 SD	G2 SD	P value
1	0.80	0.81	0.40	0.39	0.83
7	0.90	0.91	0.30	0.28	0.66
14	0.93	0.92	0.26	0.28	0.63

SD stands for 'standard deviation'. The analysis is calculated using the 275 test files. Noticing the two-tailed P value, and using conventional criteria, this difference is considered to be *not statistically significant*

manages to extract the visually important nodes from ConceptNet, whilst eliminating others.

Moreover, using all the relationships is substantially more computationally complex than using the selected subset, and thus has significantly longer execution times. It was found that this step of the experiment, with the full relationships, has taken 260 % more time for processing one video. This means that analyzing the output of Layer1 into Layer2 for each query file is about three times faster using the adapted version compared to the full ConceptNet. The full ConceptNet has also taken 6.9 times longer to be loaded into the memory, reflecting that more memory resources are needed as well.

5 Conclusion

In this paper, a new framework for automatic semantic video annotation of unconstrained videos is introduced. Besides the inherited challenges found in domain-specific video analysis, unconstrained videos lack previous information concerning their contents. They also need a huge quantity of knowledge to represent all possible scene events. This framework combines two layers: low-level visual similarity matching and annotation analysis utilizing commonsense knowledgebases.

This proposed approach provides a number of contributions: firstly, it tackles the semantic annotation of unconstrained videos; it then employs commonsense knowledgebases in the way proposed; furthermore, as the resulting annotation is meant to approximate human perception of visual scenes in order to achieve the best retrieval impact, the framework output is composed of the scene parts triplet - objects, actions and, if detected, background scenes.

To address the challenges of generality in the pre-annotated dataset, the framework is designed to handle issues such as mis-spelling, use of synonyms, and the use of free text. Text-analysis tools have been utilized to pre-process annotations, and commonsense knowledgebases have been used to obtain the wide-domain knowledge needed.

Black- and white-box testing have been performed on two challenging video datasets to study the impact of the overall framework, and to examine the effect of each layer separately. These datasets contain a considerable range of visual variations and different actions and objects. Statistical analysis and evaluation of the experimental results demonstrate that the framework can provide semantically representative annotations. Moreover, the evaluation of retrieval performance shows

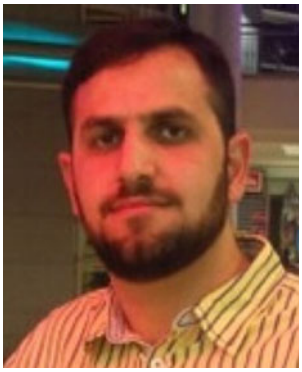
that the framework is able to retrieve the most semantically matching videos, and to rank them higher than the benchmark method.

In order to have a full annotation system, more steps are to be carried out especially in the first layer. In the first place, more complex features could be utilized. Moreover, as the computational time of Layer 1 occupies the largest percentage in processing each query video, a hierarchical approach could be applied to index selected features so as to have real-time performance. Finally, a NLP post-process stage may be plugged in to form a full meaning sentence by the addition of sentences linking parts and suitable prepositions.

References

1. Ahmed A (2009) Video representation and processing for multimedia data mining, pp 1–31. Semantic Mining Technologies for Multimedia Databases. Information Science Publishing
2. Altadmri A, Ahmed A (2009) Automatic semantic video annotation in wide domain videos based on similarity and commonsense knowledgebases. In: The IEEE international conference on signal and image processing applications, pp 74–79
3. Altadmri A, Ahmed A (2009) Video databases annotation enhancing using commonsense knowledgebases for indexing and retrieval. In: The IASTED international conference on artificial intelligence and soft computing, vol 683, pp 34–39
4. Altadmri A, Ahmed A (2009) Visualnet: commonsense knowledgebase for video and image indexing and retrieval application. In: IEEE international conference on intelligent computing and intelligent systems, vol 3, pp 636–641
5. Amir A, Basu S, Iyengar G, Lin CY, Naphade M, Smith JR, Srinivasan S, Tseng B (2004) A multimodal system for the retrieval of semantic video events. Comput Vis Image Underst 96(2):216–236
6. Bagdanov AD, Bertini M, Bimbo AD, Serra G, Torniai C (2007) Semantic annotation and retrieval of video events using multimedia ontologies. In: International conference on semantic computing, pp 713–720
7. Basharat A, Zhai Y, Shah M (2008) Content based video matching using spatiotemporal volumes. Comput Vis Image Underst 110(3):360–377
8. Bay H, Tuytelaars T, Gool LV (2006) Surf: speeded up robust features. In: European conference on computer vision, vol 3951, pp 404–417
9. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Tenth IEEE international conference on computer vision, vol 2, pp 1395–1402
10. Brox T, Malik J (2011) Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Trans Pattern Anal Mach Intell 33(3):500–513
11. Chandrasekaran B, Josephson JR, Benjamins VR (1999) What are ontologies, and why do we need them? IEEE Intell Syst Their Appl 14(1):20–26
12. Deng Y, Manjunath B (1997) Content-based search of video using color, texture, and motion. In: International conference on image processing, vol 2, pp 534–537
13. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Computer vision and pattern recognition, pp 248–255
14. Farhadi A, Hejrati M, Sadeghi M, Young P, Rashtchian C, Hockenmaier J, Forsyth D (2010) Every picture tells a story: generating sentences from images. In: The 11th European conference on computer vision, vol 6314, pp 15–29
15. Fellbaum C (1998) WordNet: an electronic lexical database. MIT Press, Cambridge, MA
16. Fergus R, Fei-Fei L, Perona P, Zisserman A (2010) Learning object categories from internet image searches. Proc IEEE 98(8):1453–1466
17. Guillaumin M, Mensink T, Verbeek J, Schmid C (2009) Tagprop: discriminative metric learning in nearest neighbor models for image auto-annotation. In: IEEE 12th international conference on computer vision, pp 309–316
18. Gupta A, Kembhavi A, Davis LS (2009) Observing human-object interactions: using spatial and functional compatibility for recognition. IEEE Trans Pattern Anal Mach Intell 31(10):1775–1789
19. Haering N, Qian RJ, Sezan MI (2000) A semantic event-detection approach and its application to detecting hunts in wildlife video. IEEE Trans Circuits Syst Video Technol 10(6):857–868

20. Hauptmann AG, Chen MY, Christel M, Lin WH, Yang J (2007) A hybrid approach to improving semantic extraction of news video. In: International conference on semantic computing, pp 79–86
21. Hsu MH, Tsai MF, Chen HH (2008) Combining wordnet and conceptnet for automatic query expansion: a learning approach. In: Asia information retrieval symposium, vol 4993, pp 213–224. Springer
22. Ikizler N, Duygulu P (2007) Human action recognition using distribution of oriented rectangular patches. In: ICCV workshop on human motion understanding, modeling, capture and animation, pp 271–284
23. Jiang YG, Yang J, Ngo CW, Hauptmann AG (2010) Representations of keypoint-based semantic concept detection: a comprehensive study. IEEE Trans Multimedia 12(1):42–53
24. Kapoor A, Grauman K, Urtasun R, Darrell T (2010) Gaussian processes for object categorization. Int J Comput Vis 88(2):169–188
25. Lenat DB (1995) Cyc: a large-scale investment in knowledge infrastructure. Commun ACM 38(11):33–38
26. Liu H, Singh P (2004) Conceptnet: a practical commonsense reasoning tool-kit. BT Technol J 22(4):211–226
27. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos in the wild. In: Computer vision and pattern recognition, pp 1996–2003
28. Lowe DG (1999) Object recognition from local scale-invariant features. In: 7th international conference on computer vision, vol 2, pp 1150–1157
29. Motulsky H (1999) Analyzing data with GraphPad prism. GraphPad Software Inc, San Diego, CA
30. Ngo CW, Jiang YG, Wei XY, Zhao W, Liu Y, Wang J, Zhu S, Chang SF (2009) Vireo/dvmm at trecvid 2009: high-level feature extraction, automatic video search, and content-based copy detection. In: TREC video retrieval evaluation workshop online proceedings
31. Niebles J, Fei-Fei L (2007) A hierarchical model of shape and appearance for human action classification. In: IEEE conference on computer vision and pattern recognition, pp 1–8
32. Over P, Awad G, Fiscus J, Antonishek B, Michel M, Smeaton AF, Kraaij W, Qunot G (2011) Trecvid 2010: an overview of the goals, tasks, data, evaluation mechanisms, and metrics. In: TRECVid 2010, pp 1–34
33. Shyu ML, Xie Z, Chen M, Chen SC (2008) Video semantic event/concept detection using a subspace-based multimedia data mining framework. IEEE Trans Multimedia 10(2):252–259
34. Siersdorfer S, Pedro JS, Sanderson M (2009) Automatic video tagging using content redundancy. In: The 32nd international ACM SIGIR conference on research and development in information retrieval, pp 395–402
35. Sivic J, Zisserman A (2009) Efficient visual search of videos cast as text retrieval. IEEE Trans Pattern Anal Mach Intell 31(4):591–606
36. Smeaton AF, Browne P (2006) A usage study of retrieval modalities for video shot retrieval. Inf Process Manag 42(5):1330–1344
37. Stanford_NLP_Group (2008) The Stanford nlp log-linear part of speech tagger (28–09–2008). http://nlp.stanford.edu/software/tagger.shtml
38. TrecVid (2011) Trec video retrieval track, bbc ruch 2005 (01–02–2011). http://www-nlpir.nist.gov/projects/trecvid/
39. UCF_Computer_Vision_lab (2011) Ucf action dataset (11–11–2011). http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html
40. Ulges A, Schulze C, Koch M, Breuel TM (2010) Learning automatic concept detectors from online video. Comput Vis Image Underst 114(4):429–438
41. Ventura C, Martos M, Nieto XG, Vilaplana V, Marques F (2012) Hierarchical navigation and visual search for video keyframe retrieval. In: The international conference on advances in multimedia modeling, pp 652–654
42. Wei XY, Jiang YG, Ngo CW (2011) Concept-driven multi-modality fusion for video search. IEEE Trans Circuits Syst Video Technol 21(1):62–73
43. Yuan P, Zhang B, Li J (2008) Semantic concept learning through massive internet video mining. In: IEEE international conference on data mining workshops, pp 847–853
44. Zhao WL, Wu X, Ngo CW (2010) On the annotation of Web videos by efficient near-duplicate search. IEEE Trans Multimedia 12(5):448–461



Amjad Altadmri received the PhD degree from the School of Computer Science at the University of Lincoln, UK in 2013. A MPhil degree in computer science from University of Lincoln, UK in 2009. A Bachelor of Engineering in Computer Science from University of Damascus, Syria in 2004. Amjad's current research focuses on Semantics of Visual contents, both from visual contents area and the link to the semantic textual one. His other research interests include Video Understanding, Ontology and Commonsense.

Amr Ahmed (BSc'93, MSc'98, PhD'04, MBCS'05, IEEE-CS'08) is a Senior Lecturer, and the Founder and the Leader of the DCAPI (Digital Contents Analysis, Production, and Interaction: http://dcapi.lincoln.ac.uk) research group at the School of Computer Science, University of Lincoln, UK. His research focuses on the analysis, understanding, and interpretation of digital contents, especially visual contents. Amr's current research interests include Contents-Based Image/Video retrieval, video and scene understanding, semantic analysis, integration of knowledge and various modalities for scene understanding.

Amr worked in the industry for several years, including Sharp Labs of Europe (SLE), Oxford (UK), as a Research Scientist, and other Engineering Consultants companies abroad. He also worked as a Research Fellow, at the University of Surrey, before joining the academic staff at the University of Lincoln in 2005.

Dr. Ahmed is a Member of the British Computer Society (MBCS) and the IEEE Computer Society. He received his Bachelor's degree in Electrical Engineering and M.Sc. degree (by research) in Computer and Systems Engineering, from Ain Shams University, Egypt, in 1993 and 1998 respectively, and his Ph.D. degree in Computer Graphics and Animation from the University of Surrey, U.K., in 2004.